

## 2.1.3.4 Hybrid of Functional and Procedural Languages

### Return to Programming Paradigm

*All programming paradigms have their benefits to both education and ability. [Functional languages](#) historically have been very notable in the world of scientific computing. Of course, taking a list of the most popular languages for scientific computing today, **it would be obvious that they are all multi-paradigm**. Object-oriented languages also have their fair share of great applications. Software development, game development, and graphics programming are all great examples of where [object-oriented programming](#) is a great approach to take.*

*The biggest note one can take from all of this information is that the future of software and [programming language](#) is multi-paradigm. It is unlikely that anyone will be creating a purely functional or object-oriented programming language anytime soon. If you ask me, this isn't such a bad thing, as there are weaknesses and strengths to every programming approach that you take, and a lot of true optimization is performing tests to see which methodology is more efficient or better than the other overall. This also puts a bigger thumbtack into the idea that everyone should know multiple languages from multiple paradigms. With the paradigms merging using the power of generics, it is never known when one might run into a programming concept from an entirely different programming language!<sup>1)</sup>*

In keeping with Boudreau conclusion, “it would be obvious that they are all multi-paradigm”, the DIDO-CLI is best served using a hybrid programming paradigm approach. Where a Functional Paradigm makes sense, it should be used and followed. Afterall, two very successful CLIs, i.e., [POSIX](#) and [Structured Query Language \(SQL\)](#), have demonstrated quite a bit of success using a Functional approach. However, both of these languages also have embraced the Procedural Paradigm also:

- SQL has Stored Procedures and triggers
- POSIX has adopted the use of scripts

In addition, Oracle RDBMS, PostgreSQL and MySQL have included support of Object Oriented Programming<sup>2)3)4)</sup>.

<sup>1)</sup>

Emmett Boudreau, [What Is A Programming Paradigm?](#), 16 October 2020, TowardsDataScience, Accessed 15 April 2021, <https://towardsdatascience.com/what-is-a-programming-paradigm-1259362673c2>

<sup>2)</sup>

TutorialsPoint, [PL/SQL - Object Oriented](#), Accessed: 15 April 2021, [https://www.tutorialspoint.com/plsql/plsql\\_object\\_oriented.htm#:~:text=PL%2FSQL%20allows%20defining%20an,and%20methods%20for%20operating%20it](https://www.tutorialspoint.com/plsql/plsql_object_oriented.htm#:~:text=PL%2FSQL%20allows%20defining%20an,and%20methods%20for%20operating%20it).

<sup>3)</sup>

tapoueh.org, [Object Relational Database Management System](#), 22 March 2018, Accessed: 15 April 2021, <https://tapoueh.org/blog/2018/03/object-relational-database-management-system/>

<sup>4)</sup>

Nick Long, [Does MYSQL Support Object Oriented Database?](https://stackoverflow.com/questions/7901907/does-mysql-support-object-oriented-database), StackOverflow, 21 August 2013, Accessed: 15 April 2021,  
<https://stackoverflow.com/questions/7901907/does-mysql-support-object-oriented-database>

From:  
<https://www.omgwiki.org/dido/> - **DIDO Wiki**

Permanent link:  
[https://www.omgwiki.org/dido/doku.php?id=dido:public:s\\_cli:05\\_contents:01\\_prt:02\\_basics:03\\_paradigim:hybrpro](https://www.omgwiki.org/dido/doku.php?id=dido:public:s_cli:05_contents:01_prt:02_basics:03_paradigim:hybrpro)

Last update: **2021/08/13 13:25**

